Low-Cost Multi-Projector Curved Screen Displays

Ramesh Raskar, Jeroen van Baar

TR2005-079   May 2005

Abstract

Current systems for high resolution curved screen displays are expensive and target high-end applications. We present a technique to create a low-cost multi-projector curved screen display using a parametric approach called the quadric transfer. This parametric approach has several advantages that are exploited to allow for flexible system configuration and the use of low-cost off-the-shelf components, yet achieve accurate alignment.

International Symposium Society for Information Displays (SID)

P.153: Low-Cost Multi-Projector Curved Screen Displays

Ramesh Raskar, Jeroen van Baar

MERL – Mitsubishi Electric Research Laboratories, Cambridge, MA, 02193, USA

Abstract

Current systems for high resolution curved screen displays are expensive and target high-end applications. We present a technique to create a low-cost multi-projector curved screen display using a parametric approach called the quadric transfer. This parametric approach has several advantages that are exploited to allow for flexible system configuration and the use of low-cost off-the-shelf components, yet achieve accurate alignment.

1. Introduction

High-resolution displays are becoming important for the ever increasing resolution demands of data visualization, and interactive simulations. Immersion plays an important role for these types of displays. Specifically curved screen displays are very suitable for the creation of immersive and collaborative environments, which we will denote as immersive Virtual Reality. This paper focuses on spherical and cylindrical displays used in immersive VR, and that use projection to display imagery on the display surface. Close proximity to these types of curved display surfaces fills the users' entire peripheral view.

Background Some curved screen displays, typically of small diameter, are using a single high-resolution projector mounted with a special lens to cover the entire display surface [1][2]. To fulfill the need for ever larger, brighter and higher resolution displays, multiple projectors in an overlapping arrangement are "stitched" together [5][6][8]. Rapid decline in size and cost of these projectors has made the multi-projector approach feasible.

In order to create a single display using multiple overlapping projectors, the projectors have to be geometrically registered. The projectors are registered with respect to another either manually or automatically using a camera-based approach. In the manual case the alignment is achieved by adjusting alignment grids through the projectors' menu. This manual alignment process is both tedious and time-consuming. In the automatic case, typically a non-parametric approach with a camera is used to determine a mapping between the camera and the projectors [8][10]. For a non-parametric approach the camera is located in the ideal viewing location or "sweet-spot" for correct perspective. However the camera location should be occupied by the user and the camera is removed once the system is calibrated. The setups are permanent and costly rigid support structures are constructed to prevent accidental shifting of the projectors.

To display images onto the curved surface using multi-projector the contents have to be distributed to multiple units where they are warped and intensity corrected to achieve alignment and seamless display. Current solutions typically use special-purpose hardware for the warping and intensity correction. The use of special-purpose hardware drives up the cost of the entire system and consequently such systems have mostly been deployed in high-end applications.

Objective In this paper we describe a flexible method using the quadric transfer described in [7] to create low-cost curved displays using multiple projectors. We discuss our overall system implementation using off-the-shelf components and show that our

Figure 1. *A projector-camera pair. The camera is attached to the projector and the pair is calibrated. The unit depicted is an example of a context-aware unit, which besides camera and projector also consists of a laptop (computing), a tilt sensor and wireless LAN.*

approach allows for flexible configurations using any desired number of projectors. We argue that our approach extends to any size screen, including wrap-around screens. Using off-the-shelf components as opposed to special-purpose hardware has two advantages: (1) leveraging off the rapid increase in components performance and (2) benefit from the large volume, decreasing cost for those components. The off-the-shelf components used for creating a multi-projector curved screen display include PCs (CPUs), networking, and graphics hardware (GPUs). Ethernet networking bandwidth is moving toward 10 Gb/sec. Advances in programmable graphics hardware allow for the programming of the vertex and pixel pipelines. We will show that our method could use very low-cost (web-)cameras.

2. Flexible Calibration

We have developed a method that specifically uses the parametric properties of the class of spherical and cylindrical surfaces: the quadric surfaces. The method is called a quadric transfer, a camera-based approach to automatically create a seamless display using multiple projectors. Using a parametric approach compared to a non-parametric approach has several advantages:

- the method is more robust to feature localization errors

- cameras are not required to be located in the ideal viewing location or "sweet-spot"

- the overlapping region between projections can be more accurately determined

- a parametric approach supports head-tracking and stereo imagery

To support flexible calibration and configurations, we attach a camera to every projector (Figure 1). By attaching a "local" camera to every projector, we eliminate the need for "global" cameras. Ideally each unit is self-contained [6] and, besides camera and projector, would include computing and networking capability. To further emphasize the use of off-the-shelf components, we have chosen for a low-cost Logitech VGA web-camera. We will first discuss the quadric transfer in section 3, followed by an explanation of our approach in section 4. We will present results in section 6, future work in section 7 and give concluding remarks section 8.

3. Quadric Transfer

Mapping between two arbitrary perspective views of an opaque quadric surface *Q* in 3D can be expressed using a quadric transfer function Ψ. The quadric transfer is analogous to a homography for transfer between 3D points on a planar surface. Both homography, as well as quadric transfer, define a transfer between pixels in homogeneous coordinates directly. While planar homography transfers can be computed from 4 or more pixel correspondences, the quadric transfer requires 9 or more correspondences [3].

The quadric transfer can be defined in a closed form using the 3D quadric surface *Q*, a 4x4 matrix for which 3D homogeneous points satisfy $X^T Q X = 0$, and additional parameters that relate perspective projection of the two views. In [9] it is shown that the quadric transfer can be defined as:

$$x' \cong Bx - \left(q^T x \pm \sqrt{\left(q^T x\right)^2 - x^T Q_{33} x} \right) e$$

In [6] it is shown that by substituting $A = B - eq^T$ and $E = qq^T - Q_{33}$, equation (1) can be reduced to

$$x' \cong Ax \pm \left(\sqrt{x^T E x} \right) e$$

A is the homography via the polar plane between the first and the second view. The equation $x^T E x = 0$ defines the outline conic of the quadric in the first view. Equation (2) shows that the quadric transfer is a homography plus a term for the quadric surface, with 16 degrees of freedom. Next we will describe a Euclidean approach for determining the quadric transfer for all projectors comprising the display.

4. Approach

Our approach is based on the observation that a camera and projector both are pin-hole devices, and thus we can apply single and multi-view computer vision techniques to determine the quadric transfer. Our goal is to achieve a flexible configuration of projectors comprising the display surface. In order to compute the quadric transfer we need to determine the following information for a given configuration:

1.) The matrix Q describing the 3D quadric surface, and

2.) The rigid transformation Γ_P^C, e.g., rotation and translation, between a camera and a projector.

We could use a temporary calibration target to compute the rigid transformations. This laborious approach is perhaps acceptable for more permanent configurations, but for our goal of flexible configurations it is not acceptable. Instead we would like to compute the rigid transformation using only display surface, projectors and cameras. We therefore created projector-camera pairs (Figure 1) by attaching a camera to each projector with some small baseline. To be able to compute rigid transformations between other projector-camera pairs, we determine the extrinsic parameters, e.g., rigid transformation $\Gamma_{P_i}^{C_i}$, for the projector-camera pair $P_i C_i$, and intrinsic parameters for both projector P_i and camera C_i. Note that the extrinsic and intrinsic parameters need to be determined only once, assuming the camera is permanently fixed to the projector, and can thus be performed in an offline process.

For *n* projector-camera pairs we can compute the quadric transfer with the following steps:
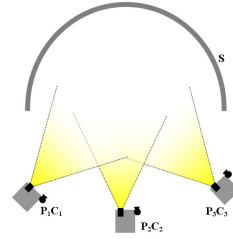
(a) Compute correspondences between cameras and

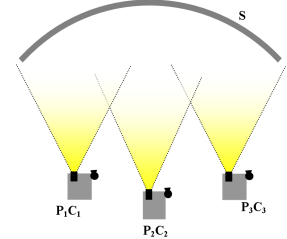Figure 2 *Example of configuration with overlap between all projections* projectors,

Figure 3 *Example of configuration with overlap only between some projections*

(b) Triangulate and find the equation of the quadric in 3D, and

(c) Compute the quadric transfer parameters.

Each projector P_i, for $i = 1..n$, projects a structured pattern on the quadric surface, one at a time and is viewed simultaneously by cameras C_i of the *n* units in the group. As discussed earlier in order to compute the quadric transfer $\Psi_{P_k}^{C_i}$, mapping the image in camera C_i to projector P_k, we first need to compute the rigid transformation $\Gamma_{P_k}^{C_i}$ between projector P_k and camera C_i. The steps to compute $\Gamma_{P_k}^{C_i}$ for a camera C_i are as follows:

1. We back-project and triangulate *m* corresponding points in P_i and C_i to obtain a set of 3D points D_i that represent the display surface, and store them in C_i coordinate system.

2. Given 3D points D_i, and 2D corresponding pixels observed by camera C_k of the pair $P_k C_k$, we can find $\Gamma_{C_k}^{C_i}$ [11]; Since $\Gamma_{P_k}^{C_k}$ is known from offline calibration, then $\Gamma_{P_k}^{C_i}$ is found from

$$\Gamma_{P_k}^{C_i} = \Gamma_{P_k}^{C_k} \Gamma_{C_k}^{C_i}.$$

3. Fit a 3D quadric Q_i to points D_i by solving the system $D_{i,j} Q_i D_{i,j} = 0$ with $j = 1..m$.

4. Construct projection matrix M_{P_k} from $\Gamma_{P_k}^{C_i}$. Then find linear solution parameters of $\Psi_{P_k}^{C_i}$ from Q_i, M_{C_i} and M_{P_k}

5. Perform nonlinear refinement of $\Psi_{P_k}^{C_i}$ parameters by minimizing pixel re-projection errors, i.e., distance between projector pixels $p_{k,j}$ and $p'_{k,j}$, obtained from quadric transfer of features from camera C_i.

Once we determined the quadric transfer $\Psi_{P_k}^{C_i}$ we can use it to pre-warp the images before projection. As we will explain later we also compute the "inverse" of $\Psi_{P_k}^{C_i}$, i.e., $\Psi_{C_i}^{P_k}$, in step 5. Note that this inverse $\Psi_{C_i}^{P_k}$ is computed in the same manner as $\Psi_{P_k}^{C_i}$, since we cannot directly compute $(\Psi_{P_k}^{C_i})^{-1}$. The linear solution calculated in step 4 results in relatively large re-projection errors, which manifest themselves visually as misregistration between projected images. The large re-projection errors are the result of accumulating errors in the steps prior to computing the linear solution. Especially finding the pose of a camera from known 3D points on a quadric is error-prone because the 3D points are usually near-planar. The iterative approach from [11] reduces this error somewhat. To ensure accurate registration between the projected images, i.e., reduce the re-projection errors, we need non-linear refinement of the quadric transfer parameters as outlined in step 5. The non-linear refinement increases the time

required to perform the system calibration and hence the calibration process is not real-time, but on the other hand the calibration procedure is completely automatic.

The outlined approach demonstrates that the technique is independent of the number of projectors that make up the display. The only requirement is that projections have sufficient overlap with neighboring projections. The approach works best for cases where all the projected images overlap one another. Consider for example the configuration depicted in Figure 2. Here we use three projector-camera pairs to render an image on a curved display **S**. Since all three projections overlap sufficiently, we can compute the quadric transfer between each projector and camera directly. If we for example choose the camera $\mathbf{C_1}$ to be our origin, we can compute the quadric transfers $\Psi_{P_1}^{C_1}$, $\Psi_{P_2}^{C_1}$, and $\Psi_{P_3}^{C_1}$ (and the inverses) for the projectors $\mathbf{P_1}$, $\mathbf{P_2}$, and $\mathbf{P_3}$. On the other hand consider the case of Figure 3. In this configuration there is virtually no overlap between the projections of projector $\mathbf{P_1}$ and projector $\mathbf{P_3}$, and consequently we cannot compute a quadric transfer between these projector-camera pairs. However, as we will explain in the next section, for the purpose of rendering using the quadric transfer, we do need a quadric transfer between the pairs. By concatenating known quadric transfers, we can compute $\Psi_{P_3}^{C_1}$ as $\Psi_{P_2}^{C_1}\left(\Psi_{C_3}^{P_2}\left(\Psi_{P_3}^{C_3}\right)\right)$. Concatenating quadric transfers aggregates errors depending on the order of concatenation. This is because $\Psi_{P_2}^{C_1}\left(\Psi_{C_3}^{P_2}\left(\Psi_{P_3}^{C_3}\right)\right)$ typically has slightly different error from $\Psi_{P_3}^{C_1}$, assuming we were able to determine $\Psi_{P_3}^{C_1}$ directly. To solve this problem we should compute the quadric transfers on a global solution basis, as opposed to the current local solutions. However, this remains future work.
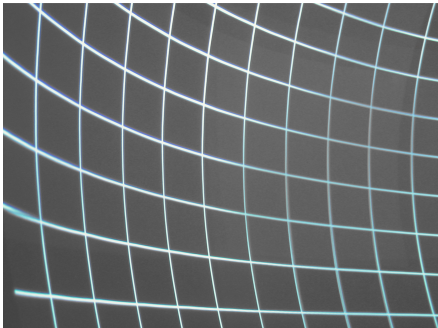

Figure 4 *Registration quality for a four projector setup on a 1.5m diameter dome screen*

Figure 4 shows an example of the accuracy of our approach. The configuration consists of four overlapping projectors on 1.5 meter dome, and we can compute the quadric transfers between all projectors and cameras directly. We have achieved similar accuracy on a 2 meter diameter dome as well.

6. Rendering and Results

For curved surfaces, the imagery is typically rendered in a two-pass approach: in a first step the image for a given viewing location is formed, and then in a second step the images are pre-warped for each projector (using a look-up table). The two-pass approach requires image resampling which degrades the final image quality. A single pass approach does not require a resampling step. Nowadays off-the-shelf graphics hardware allows certain parts of the graphics pipeline to be programmed via so-called vertex and pixel (fragment) shaders. We have implemented the pre-warping of the input data in a single-pass vertex shader.

For a given viewpoint we can pre-warp the input images with the following steps. Given a 3D vertex *M* in the scene to be rendered, we find its screen space coordinates *m* in the virtual view by projecting *M* using the intrinsic parameters of a real camera. To ensure correct visibility we save depth information for *m*. Next, we apply the quadric transfer Ψ_P^C to *m* to find the transferred pixel coordinate *m'* in the framebuffer of the projector. The polygons in the scene are then rendered with vertices *M* replaced with vertices *m'* (and saved depth information inserted). Since we pre-warp geometry rather than images, we achieve a higher quality rendering for curved displays. After geometry pre-warping we rely on OpenGL for rasterization.

The rendering process for each projector is the same. Each projector framebuffer automatically picks up the appropriate part of the image, and during the calibration step we have determined a quadric transfer between the current viewing location and every projector that comprises the display. There is no need to explicitly figure out the extents of the projectors. To illustrate our method, we have implemented the rendering using quadric transfer in Chromium [4]. Each projector is connected to a client PC in the cluster. A server PC distributes the OpenGL commands to the clients. The server PC sends each client the correct quadric transfer parameters, or the parameters of multiple quadric transfers if concatenated, and each client renders the geometry using the provided parameters. Figure 5 shows a screenshot of a scene rendered using Chromium and the vertex shader.

Rendering the entire scene for every projector in the display could quickly result in network bandwidth problems. Although a discussion of distributed rendering is beyond the scope of this paper, we can use the quadric transfer to determine the visible primitives per projector. This process, known as geometry bucketing, aims to reduce the network load by reducing the data that is sent over the network. Because the pre-warping calculations using the quadric transfer are implemented in a vertex shader, all pre-warping occurs on the graphics processing unit (GPU), the free cycles of the CPU can thus be used for geometry bucketing.

Since we pre-warp the geometry in the vertex shader, as opposed to pixels, and scan-conversion in OpenGL linearly interpolates the information between the vertices, large primitives exhibit mis-registration along their edges. The edges between vertices should map to second degree curves according to the underlying surface. To avoid this problem, we need to reduce the size of the underlying primitives. We have implemented a straightforward dynamic tessellation approach that tessellates primitives according to their projected size.

Intensity Weighting for Overlapping Regions

As is evident from the top image of Figure 5, in areas of projection overlap, the intensity is brighter compared to areas of non-overlap. In order to make the display appear seamless, we need to adjust, or weight, the intensity for each projector in the overlap areas. To determine the weighting for a given pixel in a projector, we need to determine which pixels from other projectors also contribute to that pixel location. We can use $\Psi_{P_k}^{C_i}$ and its inverse $\Psi_{C_i}^{P_k}$ to determine those projector pixels in the other projectors. The intensity correction is computed after the calibration and stored as a monochrome image, which is subsequently used as an alpha map during rendering. The result is shown in the bottom image of Figure 5.
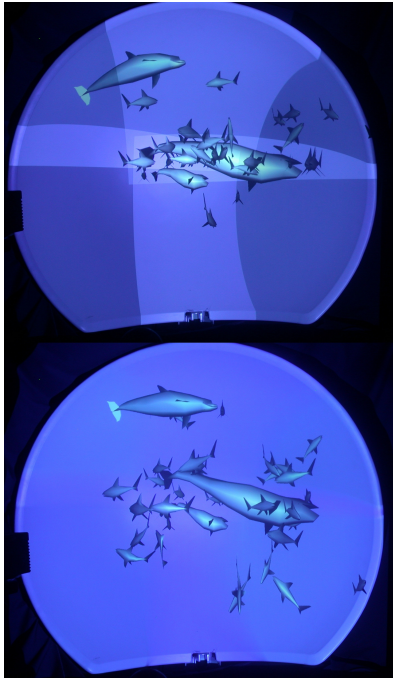
P.

Figure 5 *Two screenshots of rendering application. Rendering without intensity correction (top) and rendering with intensity correction (bottom)*

Pre-recorded Video Stream Playback

In the previous section we only discussed real-time rendering of 3D geometry. Although not in real-time, we could also display pre-recorded video streams using the quadric transfer. Each frame of the pre-recorded video can be pre-warped off-line and after all pre-warped frames have been processed, they can once again be collected into a video stream for playback.

7. Impact

We have demonstrated a technique which can be used to create multi-projector imagery on curved surfaces. Although our approach is limited to surfaces that adhere to the class of quadric surfaces, as opposed to [5] and [8], we exploit the advantages of a parametric approach. We have achieved pixel accurate registration on hemispherical displays of 1.5 and 2 meter in diameter.

Our technique exploits the usage of off-the-shelf components to reduce the cost of the system. An advantage of off-the-shelf components is that we can leverage off their increasing performance. We can easily replace older components with higher performance components. Our flexible technique thus allows for a heterogeneous setup, i.e., different brand and type components can be mixed to build the multi-projector display cluster.

So far we have not dealt with the issue of photometric correction between the projectors in a multi-projector display. For a homogeneous setup and approximately same intensity light bulbs, the difference in color and intensity among projectors is often minimal. However for a heterogeneous setup, or different intensity light bulbs, the difference in color and intensity between projectors could have a negative impact on the perception of a seamless display.

Two important aspects of immersive VR are rendering stereo imagery and head-tracking. With the prospect of ever increasing performance for off-the-shelf components, stereo imagery at interactive frame-rates will become feasible in the near future.

8. Conclusion

We have presented a technique to accurately register multiple projectors on a curved screen display. Using projector-camera pairs, the approach we outlined is flexible with respect to the configuration of projectors. Installation and maintenance are low-cost. It also allows the use of off-the-shelf components to keep overall system cost low. This makes the application of curved screen displays feasible in areas such as entertainment, CAD design, and education. One of the advantages of the quadric transfer, .i.e., a parametric approach, to pre-warp the input, is that it allows us to position the cameras anywhere, rather than having to put them in the intended viewing location as is the case for non-parametric techniques.

Acknowledgements

References

[1] Biehlig W., Deter C., Dube S., Hill B., Helling S., Isakovic K., Klose S., Schiewe M.: *LaserCave – Some Building Blocks for Immersive Screens*. 2003.

[2] Elumens Corp., http://www.elumens.com

[3] Forsyth, Ponce, Computer Vision, A Modern Approach. 2002.

[4] Humphreys G., Houston M., Ng Y., Frank R., Ahern S., Kirchner P., Klosowski J.: Chromium: *A stream processing framework for interactive graphics on clusters.* In Proceedings of SIGGRAPH 2002 (2002).

[5] Jaynes C., Webb S., Steele R.M., *A Scalable Framework for High-Resolution Immersive Displays*, University of Kentucky Technical Report.

[6] Raskar R., van Baar J., Beardsley P., Willwacher T., Rao S., Forlines C.: ilamps: *Geometrically aware and self-configuring projectors*. ACM Trans. Graph. 22, 3 (2003), 809.818.

[7] Raskar R., van Baar J., Willwacher T., Rao S.: Quadric Transfer for Immersive Curved Screen Display, Computer Graphics Forum, Volume 23, Issue 3, Page 451 September 2004.

[8] Raskar R., Welch G., Fuchs H.: *Seamless Projection Overlaps Using Image Warping and Intensity Blending*. In Fourth International Conference on Virtual Systems and Multimedia (Gifu, Japan, Nov. 1998).

[9] Shashua A., Toelg S.: *The quadric reference surface: Theory and applications*. International Journal of Computer Vision (IJCV) 23, 2 (1997), 185.198.

[10] Trimension Systems Ltd., http://www.trimension-inc.com/, Dec 2002.

[11] Lu C.P., Hager G. D., Mjolsness E.: *Fast and globally convergent pose estimation from video images*. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 6 (2000), 610.622